

**Amendments to the Claims:**

This listing of the claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A method for determining the operational characteristics of a program, comprising a verification procedure comprising the following steps:

- a first step comprising:

- expressing the operational characteristics of the program as functions dealing with occurrences or sequences of occurrences of events ~~which may occur~~ occurring during ~~possible~~ executions of the program, said events being able to deal with particular operations, particular values of data, at particular program points and in particular states of the program;
- determining a ~~possible~~ level of precision with which these characteristics must be determined;
- determining a ~~possible~~ set of particular contexts of execution in which the program will always be executed; and
- determining ~~possible~~ operational specificities of a set of platforms on which the program will be executed;

- a second step of estimation, by program analysis, and in consideration of said ~~possible~~ level of precision, of said ~~possible~~ set of particular contexts of execution and of said ~~possible~~ operational specificities of platforms, of information relating to ~~[[the]]~~ a structure of the program, ~~the possible~~ execution paths of the program and to ~~[[the]]~~ values of ~~possible~~

data, at various points of the execution paths and under different execution conditions, of  
[[the]] states of the program and data handled by the program; and

- a third step for determining said operational characteristics, by means of the  
information extracted by said program analysis, by [[the]] computation of said functions on  
the occurrences or particular sequences of occurrences of particular operations, dealing with  
particular values, at particular points of the program, in particular states of the program, for  
the set of execution paths determined by analysis.

2. (Currently Amended) The method according to claim 1, wherein, in the case  
when the program is interactive and may depend on an undetermined number of dynamic  
values resulting from this interaction, the contexts of execution are given by a description  
abstracted from ~~possible~~ series of data representing said dynamic values.

3. (Currently Amended) The method according to claim 1, wherein, in the  
case where the program is inserted into a framework of execution, said second step of  
estimation comprises static analysis which also take into account the semantics of this  
framework of execution, including ~~possible~~ implicit interaction loops of the program.

4. (Currently Amended) The method according to claim 1, wherein certain of  
said particular operations, ~~(which~~ which form events, accompanied by constraints on the  
values handled, the execution points, and the statuses of the ~~program)~~ program, are defined as  
one of the following actions: call to a given routine, access to a given variable, reading or  
writing on a given port, computation of a given arithmetic expression, completion of

execution of the program or of a routine ~~(on-on~~ a normal return or ending an  
~~exception)~~exception.

5. (Currently Amended) The method according to claim 3, wherein certain of said static analysis consist of abstract interpretations of the program, on abstract domains which may notably represent ~~possible~~-sets of values and symbolic expressions.

6. (Previously Presented) The method as claimed in claim 1, wherein said extracted information are represented by means of one or more of the following structures: status graph of the program, inheritance graph, graph of the routine calls of the program, control flow chart of each routine of the program, structure of loops and catch-up of exceptions, structure of basic blocks, abstraction of the status of the program at an execution point.

7. (Previously Presented) The method according to claim 1, wherein said extraction of information does not apply to unnecessary information for determining the operational characteristics, both from the viewpoint of the amount of information extracted and from the precision of these pieces of information.

8. (Previously Presented) The method according to claim 1, wherein only major pieces of information among said extracted information are computed and saved and in that the other pieces of information are only computed when necessary for determining said operational characteristics.

9. (Currently amended) The method according to claim 8, wherein the major pieces of information are information extracted at breakdown nodes of the code of routines in a graph of basic blocks and in that the other pieces of information, ~~(in~~in the body of the basic ~~blocks)~~blocks, are recomputed by local analysis from information saved at the start and end of the corresponding block.

10. (Currently Amended) The method according to claim 1, wherein said operational characteristics represent validity criteria and in that said determination establishes that the program is valid, ~~(because~~because it observes each of said ~~criteria)~~criteria, or invalid, ~~(because~~because at least one of said criteria cannot be ~~observed)~~observed.

11. (Previously Presented) The method according to claim 10, wherein said validity criteria express security or interoperability rules.

12. (Previously Presented) The method according to claim 1, wherein said operational characteristics characterise resources which are consumed and functionalities which are exploited by the program during its execution and in that said determination provides an execution profile of the program.

13. (Previously Presented) The method according to claim 3, wherein a computation of certain of said functions associated with the operational characteristics is performed during said static program analysis, as soon as certain of said pieces of information are extracted.

14. (Currently Amended) ~~Application of the~~ The method according to claim 10 for automatic filtering of a set of programs relative to a given set of validity criteria, wherein the extraction of information by static program analysis is only completed once per program and reused whenever necessary for determining whether the program observes said set of validity criteria.

15. (Currently Amended) A ~~system~~ method for distribution of applications ensuring that the applications observe validity criteria associated with the execution platforms of these applications, comprising filtering ~~means designed~~ step such that, for any client desiring to accede to the applications for a certain execution platform, the applications are filtered by a verification procedure in accordance with the method according to any one of claims 1 to 12, only the applications which observe the validity criteria for said platform being presented to the client.

16. (Currently Amended) A system for multi-application execution ensuring that the applications observe given validity criteria, comprising:

- an application analysis server, a server for validation of applications and a multi-application platform, and
- means for ensuring, prior to loading or execution of an application on the platform:
  - observance by this application of said validity criteria an extraction of information being carried out on the application analysis server and an evaluation of said validity criteria being carried out on the server for validation of applications, and

- in the case when one of the validity criteria cannot be observed, a failure of loading or execution of the application, a change of the status of the system and an emission of a sound or visual signal to alert of failure of loading or execution,

the means for ensuring observance by said application of said validity criteria executing a procedure comprising the following steps:

- a first step comprising:

- expressing the validity criteria of the program as functions dealing with occurrences or sequences of occurrences of events ~~which may occur~~ occurring during ~~possible~~ executions of the program, said events being able to deal with particular operations, particular values of data, at particular program points and in particular states of the program;
- determining a ~~possible~~ level of precision with which these validity criteria must be determined;
- determining a ~~possible~~ set of particular contexts of execution in which the program will always be executed;
- determining ~~possible~~ operational specificities of a set of platforms on which the program will be executed;

- a second step of estimation, by program analysis, and in consideration of said ~~possible~~ level of precision, of said possible set of particular contexts of execution and of said ~~possible~~ operational specificities of platforms, of information relating to [[the]] structure of the program, ~~the possible~~ execution paths of the program and to the values of possible data, at various

points of the execution paths and under different execution conditions, of

[[the]] states of the program and data handled by the program;

- a third step for determining said validity criteria, by means of the information extracted by said program analysis, by [[the]] computation of said functions on the occurrences or particular sequences of occurrences of particular operations, dealing with particular values, at particular points of the program, in particular states of the program, for the set of execution paths determined by analysis.

17. (Previously Presented) The system according to claim 16, wherein the server for validation of applications is executed on the multi-application platform, the application analysis server executing outside the platform.

18. (Previously Presented) The system according to claim 16, wherein the application analysis server and the server for validation of applications are executed on the multi-application platform.